

Falcom eCos SDK

Installation instructions

Index of contents

0	OVERVIEW	3
0.1	RELATED DOCUMENTS.....	4
1	MINIMUM SYSTEM REQUIREMENTS	5
2	INSTALLING THE SDK	6
3	CONTENTS OF THE ECOS SDK DIRECTORY	8
4	START WITH DEMO APPLICATIONS	9
4.1	START HELLO PROJECT.....	9
5	HOW TO DOWNLOAD GENERATED PROJECTS INTO THE FLASH OF TARGET DEVICE.....	14
6	APPENDIX	17
6.1	BRIEF DESCRIPTION OF DEMO PROJECTS	17
6.1.1	Demo Hello.....	17
6.1.2	Demo Thread	17
6.1.3	Demo Two-thread.....	17
6.1.4	Demo Serial echo.....	17
6.1.5	Demo Serial Bridge	17
6.1.6	Demo IOs.....	18
6.1.7	Demo Flash.....	18
6.1.8	Demo Http client.....	18
6.1.9	Demo Http Server.....	18
6.2	ISP, GPRS CONFIGURATION PARAMETERS OF GERMAN SERVICE PROVIDER	19

Version history:

Version number	Author	Changes
1.00	Fadil Beqiri	Initial version
1.01	Fadil Beqiri	The steps described in Chapter 4.1 changed (updated).

Cautions

Information furnished herein by FALCOM is believed to be accurate and reliable. However, no responsibility is assumed for its use. Also the information contained herein is subject to change without notice.

Please, read carefully the safety precautions.

If you have any technical questions regarding this document or the product described in it, please contact your vendor.

General information about FALCOM and its range of products is available at the following internet address: <http://www.falcom.de/>

Trademarks

Some mentioned products are registered trademarks of their respective companies.

Copyright

The Falcom eCos SDK description is copyrighted by FALCOM GmbH with all rights reserved. No part of this user's guide may be produced in any form without the prior written permission of FALCOM GmbH.

FALCOM GmbH.

No patent liability is assumed with respect to the use of the information contained herein.

0 Overview

Falcom eCos SDK is a free Development Kit and an real-time operating system targeted to embedded applications for Falcom Devices. This Development Kit supports a powerful and opensource integrated development environment (IDE) eClipse. The eCos SDK contents the eCos Repository for our Hardware in combines with GNU Toolchain.

The Falcom eCos SDK is a complete solution for development, highly integrated, network-enabled embedded designs.

The core eCos system consists of a number of different components such as the kernel, the C library, and an infrastructure package. Each of these provides a large number of configuration options, allowing application developers to build a system that matches the requirements of their particular application.

The Falcom eCos SDK is designed to support real-time applications by providing features such as different methods for synchronization, and a selectable scheduling policy. Device drivers, memory management, exception handling, timers, counters, and standard C and math libraries are all available. Complete development and debug tools are also provided, including software configuration and build tools, GNU-based compilers, assemblers, linkers, debuggers, and simulators.

The eCos SDK distribution is available for Windows versions. The Windows version has been tested under Microsoft Windows NT4.0, Windows 2000 Professional and Windows XP.

The concept behind this approach is to provide free sources and quickly develop a variety of highly customized wireless applications in a fastest way from product concept to market entry.

The Falcom eCos SDK offers:

- Opensource, no license cost
- real-time operating system targeted to embedded applications
- All code in source, can be changed, added and modified to the user requirements.
- Opensource community for FAQ and Problems
- ...

0.1 Related documents

Please refer to the “[universal_evaluation_board_manual.pdf](#)” user manual.

1 Minimum System Requirements

The hardware and software requirements to use eCos SDK are as follow:

- Standard Intel architecture PC running Microsoft Windows NT (no earlier than 4.0 + SP6a), Windows 2000 and Windows XP.
- 350 Mbytes free disk space or more for the installed distribution.
- 256MB of RAM and a 500MHz or faster Pentium processor.
- Free available RS232 serial port which enables the communication to the target device, to update your own developed application the on-board Flash-Memory (or testing the included demo projects released from Falcom).

Hint: The Falcom eCos SDK requires the Java 2 Runtime Environment (JRE) version 1.4 or later that has to be installed before you running the eCos SDK. The JRE is copyrighted by Sun Microsystems, Inc., and is available on the Sun Microsystems Java home page at <http://java.sun.com/>.

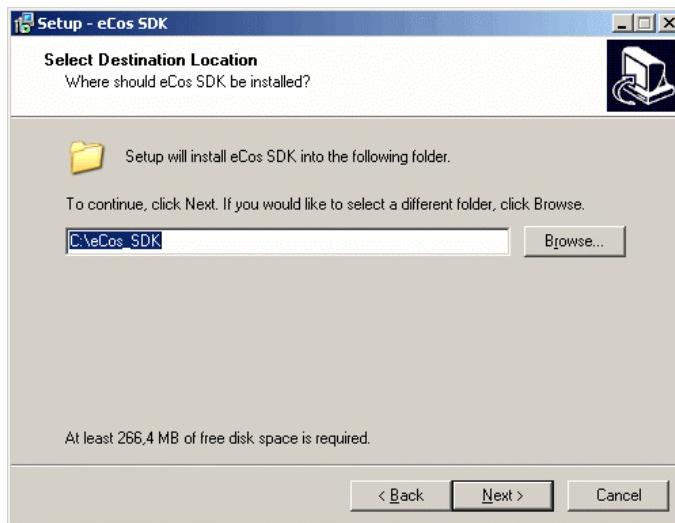
2 Installing the SDK

To install the eCos SDK insert the CD into the CD drive and follow step-by-step instructions below:

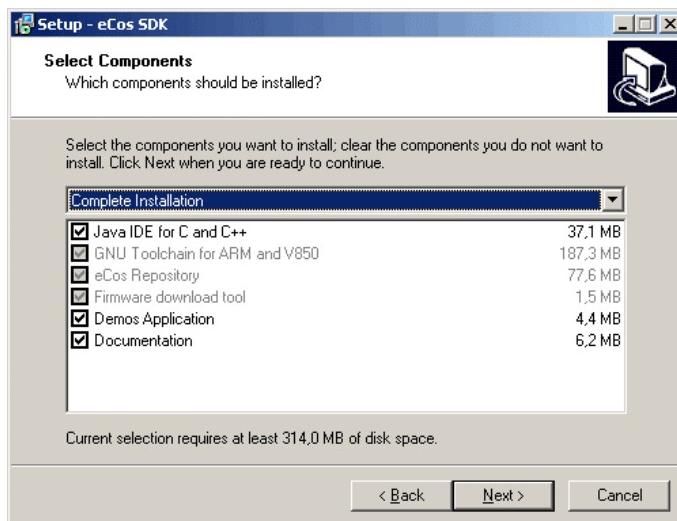
1. Make sure that you have removed any previous installation of eCos and Java Runtime Environment that are already installed. Then run the installation program of software distributions.
2. Run `\win32\ecos_sdk_setup.exe` by double click on its icon (eClips IDE, GNU toolchain, eCos repository, Documentation, Demos etc. are also included in this file and will be saved under the installation directory). On the next appeared window click the **Next** button.



3. Select the destination directory by clicking the **Browse** button where the eCos SDK should be installed.



4. Select listed components, which you need.



5. To complete the installation is recommended to restart your system (PC).



3 Contents of the eCos SDK Directory

Follow all tools and manuals included into the eCos SDK directory are listed.

- ***Doc (Documentation for eCos)***

The complete software documentation set is included under the eCos SDK directory.

- ***Ecos2 (eCos Repository)***

eCos Repository is a directory structure where all the packages get installed. The component framework comes with an administration tool that allows new packages or new versions of a package to be installed, old packages to be removed, and so on. The component repository includes a simple database, maintained by the administration tool, which contains details of the various packages.

- ***Eclipse (Java IDE for C and C++)***

Eclipse is a kind of universal tool platform - an open extensible IDE for anything and nothing in particular. It provides a feature-rich development environment that allows the developer to efficiently create tools that integrate seamlessly into the Eclipse Platform.

- ***Flashtool (SiRF firmware downloader)***

This software allows users of Falcom eCos SDK to update the demo or own developed software on-board Flash-Memory of the target devices.

- ***H-i686-pc-cygwin (Toolchain with gcc-3.3.2)***

The GCC (GNU C-Compiler) is the GNU Compiler Collection, which currently contains front ends for C, C++, Objective-C, etc. as well as libraries for these languages.

4 Start with Demo Applications

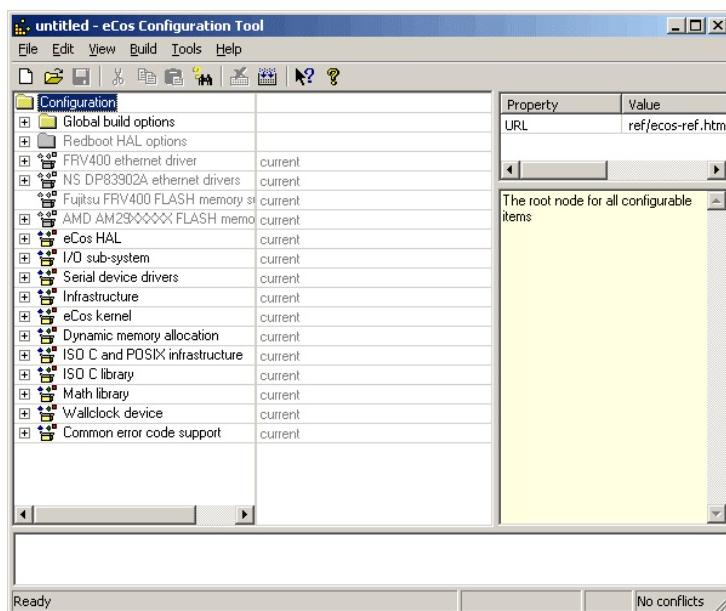
When a new project is created, the developer has an option to follow pre-made templates or create a new template. The Demo applications contain a working directory structure and make files. The IDE creates a new project with the chosen application.

The Falcom universal evaluation board is designed to provide flexibility for evaluation and extensibility for development work. The Falcom universal evaluation board includes LEDs, switches, buttons and all the I/O pins accessible to aid in quick evaluation. The universal evaluation board connects the target device to the system programming port to enable device programming and testing.

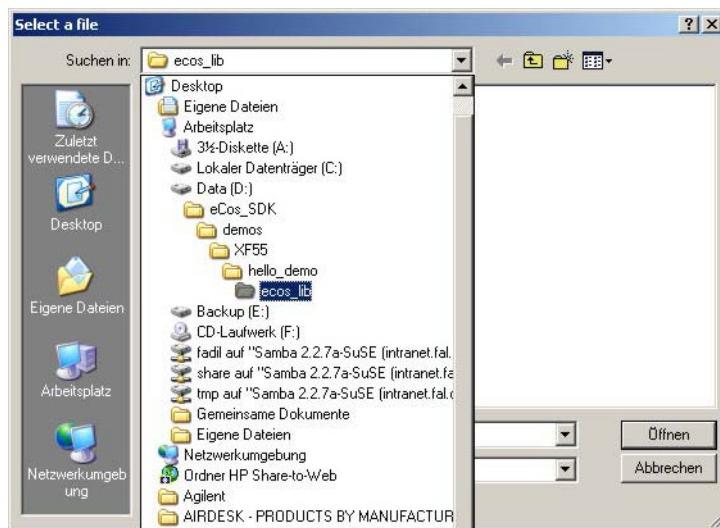
This chapter demonstrates how to start a new project by copying one of the existing released demo projects. As example is used the *Hello* project. The *Hello* project is a simply project which delivers a message to the serial port of target device. The message is displayed on a terminal program (e.g. HyperTerminal).

4.1 Start Hello project

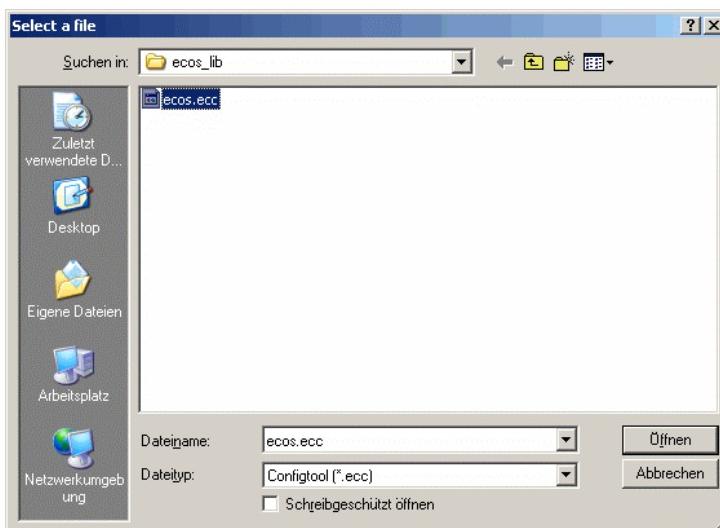
1. Start eCos Configuration Tool from the installation directory or from the desktop (if a linkage is created) by double click on the “eCos Configtool.exe” icon and open the config file for Hello project, see figure below. Go to main menu, click the File menu and start the Open program.



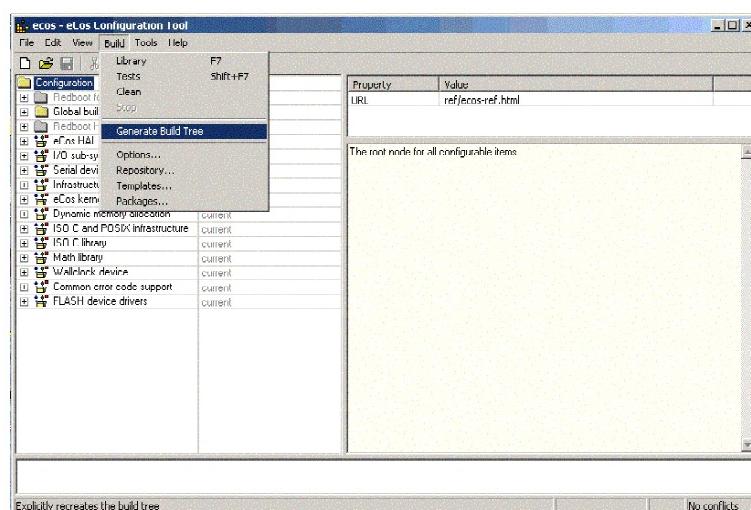
2. Select the `ecos_ib` subdirectory.



3. Chose the “ecos.ecc” file and click the Open button.

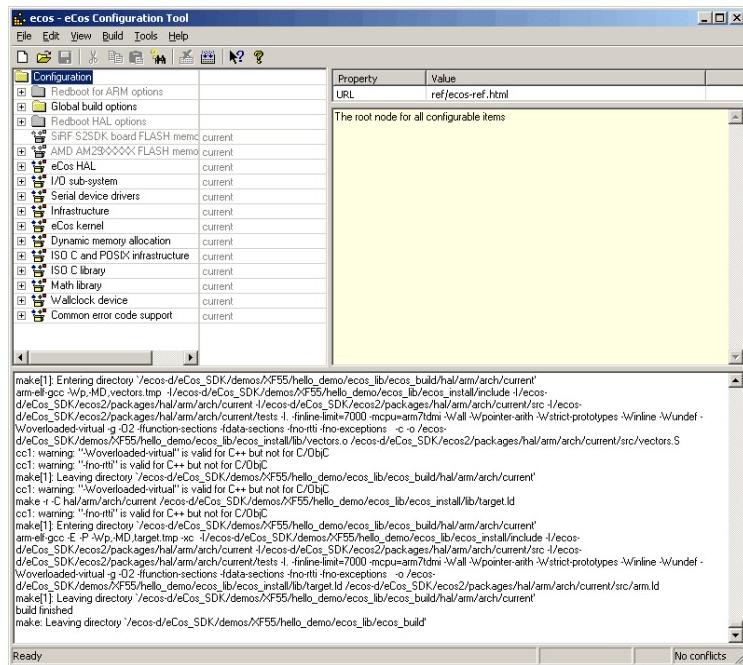


4. Go to main menu, click the Build menu and start the Generate build tree program.

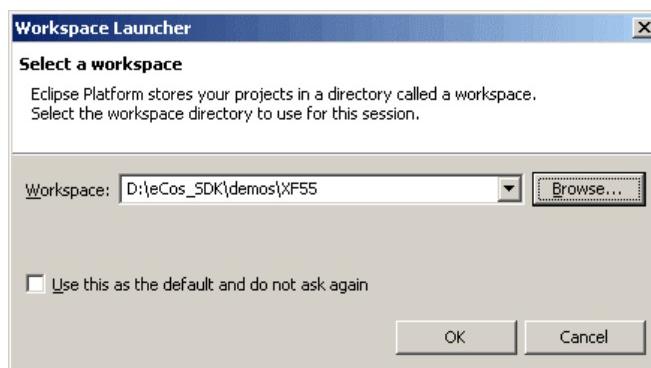


5. Go to main menu, click the Build menu and start the Library program for Hello project. If the Hello project is successfully built then follow

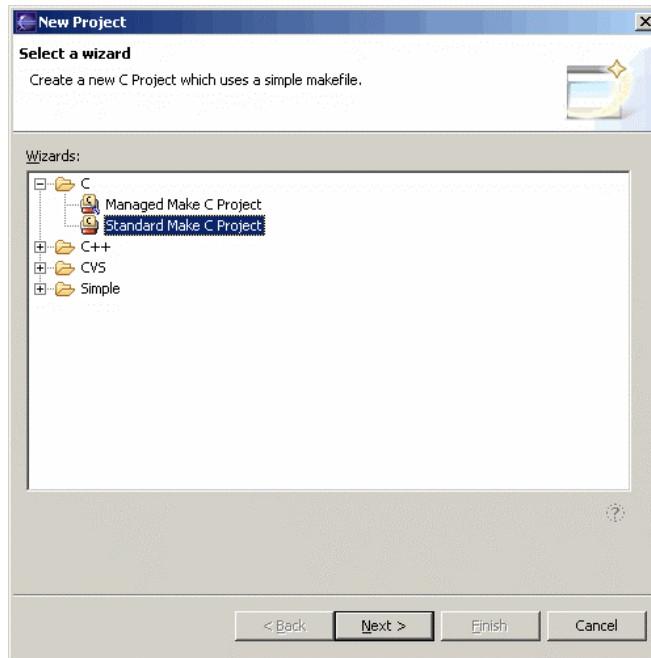
the next step, else if the output window (on the bottom of eCos configuration tool) displays any error, please restart the program and try it again.



- Start eClipse from the installation directory or from the desktop (if a linkage is created) by double click on the ..\eClipse 3.0.exe icon and set the workspace into the demos Directory, (see figure below).

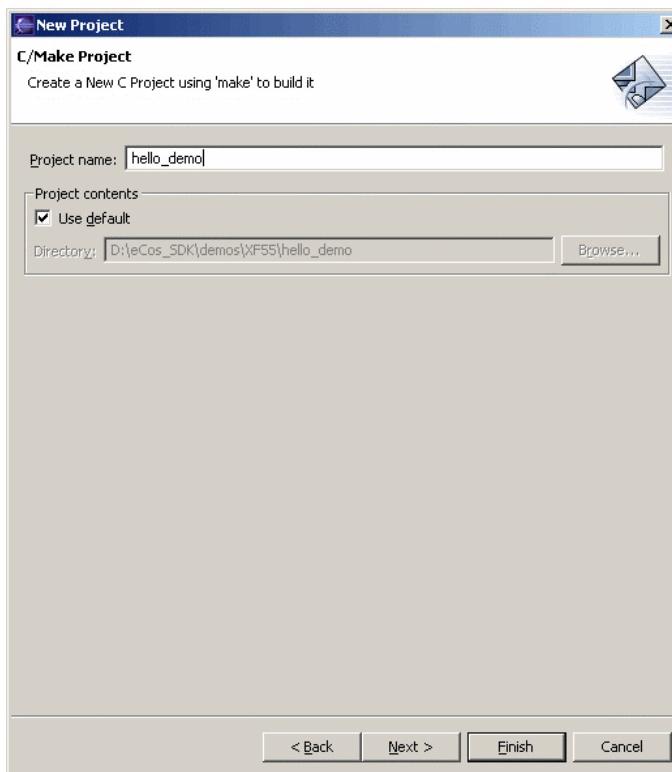


- Create New eClipse project for Hello Demo. On the appeared window choose marked item “Standard Make C Project”, as shown below, then click Next button.

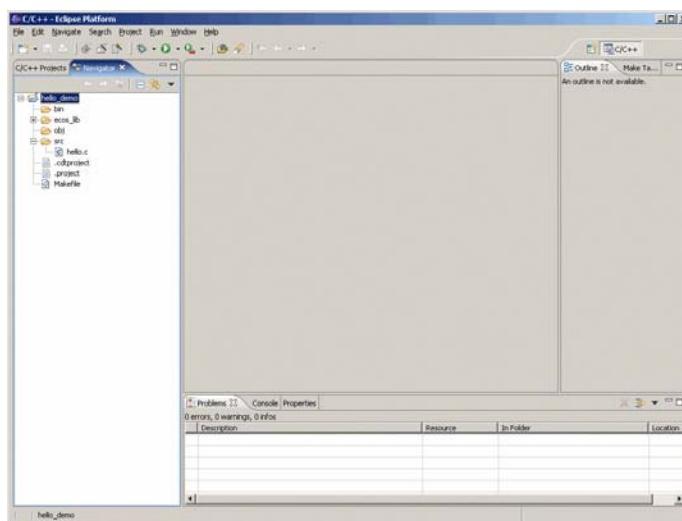


8. On the appeared screen assign the project name for the current project ("hello_demo") and click Next. On the next appeared screen just click the **Finish** button.

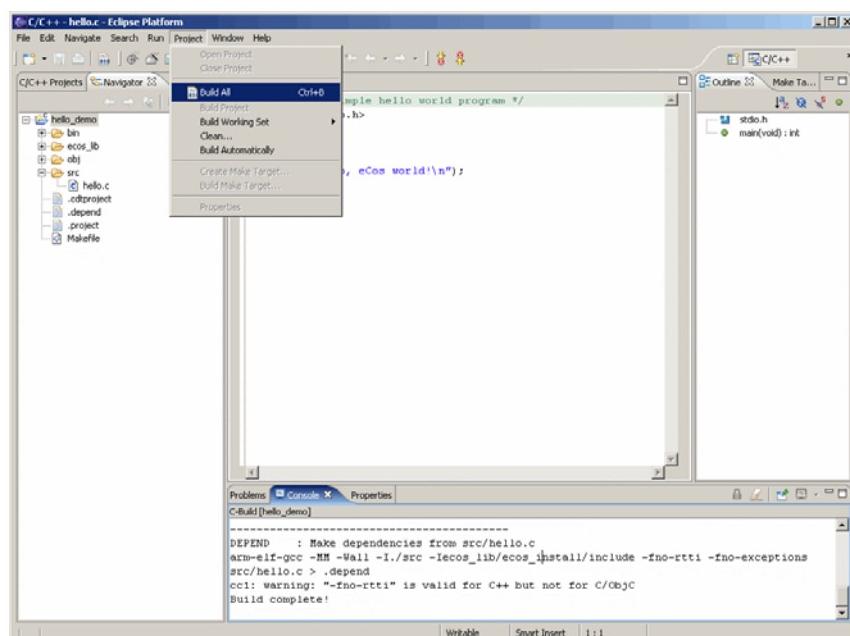
Note that the project name on this field must be the same one to the folder name of current project (e.g. "flash_demo" for flash_demo project, see folder name on the eCos_SKD directory), else the new project could not find the files allocated in this subdirectory.



9. If a message box prompts you to switch perspectives, click **Yes**. Projects files and elements are displayed in the C/C++ Projects tab and in the Navigator tab.



10. In order to have a view of the source code of the current project, open folder “*src*” on the “*C/C++ projects*” tab, then choose the included “*hello.c*” file, it opens in the editor view. If the compiling of the *Hello* demo project is not automatically performed, then go to main menu, click the *Project* menu and start the *Build all* program. Eclipse then calls the development toolchain to compile, assemble, link project files, and produce an executable file in binary format.



If compiling has successfully been completed, follow the next chapter “[Downloading compiled project into the Flash memory of target device](#)”.

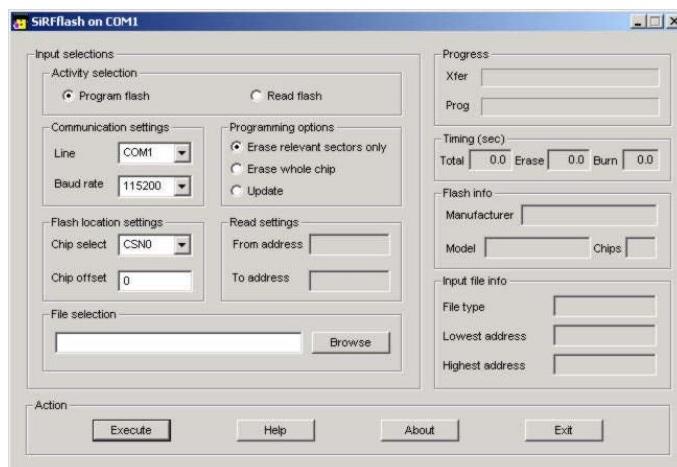
5 How to download generated projects in the FLASH of target device.

In order to allow users of Falcom eCos SDK to utilize their own generated projects into the FLASH of targed device, a program called *SirfFlash* is available, and can be accessed from the eCos SDK subdirectory. Created folder name is SirfFlash and execute file is SiRFFlash.exe.

The upload procedure described below is focused on the Hallo demo application. However, in the same way the instructions included below have to be performed for other applications.

In order to avoid any mistakes during the upload procedure, please follow step-by-step the instructions described below:

- If target device is still not connected to its corresponded evaluation board, connect it (follow the instructions described on the target device evaluation board, see [1]), else go to the next step.
- The download procedure takes place if the target system is changed from normal mode into the internal boot mode or it is powered in the boot mode (follow the instructions described on the target device evaluation board, see [1]).
- Make sure the serial port of evaluation board target device is properly connected to one of the available and unoccupied COM ports on your PC (i.e. COM1) and the target system is already set into the boot mode.
- Please refer to the figure below for the following points. Select the Program flash radio button in the Activity selection box
- Select the Line where your target device is connected (i.e COM1) and Baud rate (115200) in the Communication settings box.
- Select the Erase relevant sectors only option, in the Programming options box.
- Select the Chip select (CSN0) and Chip offset (0) in the Flash location settings box.



- Select the input file (executable compiled file “*.bin” extention, e.g. “hello.bin”) by typing the filename into the File selection field or use

the Browse button to select the required file. The file directory is shown in the figure below, then press **Open** button. The File type has to chosen to *.bin.



- Press Execute button to start the flash update programming.
- The current positions of the programming progress will be displayed in the Progress bar and the Total, Erase and Burn process will be shown in the Timing box.

Note: Do not interrupt this procedure. A partially reprogrammed flash memory could lead to invalid operation of the terminal and permanent damage to the components.

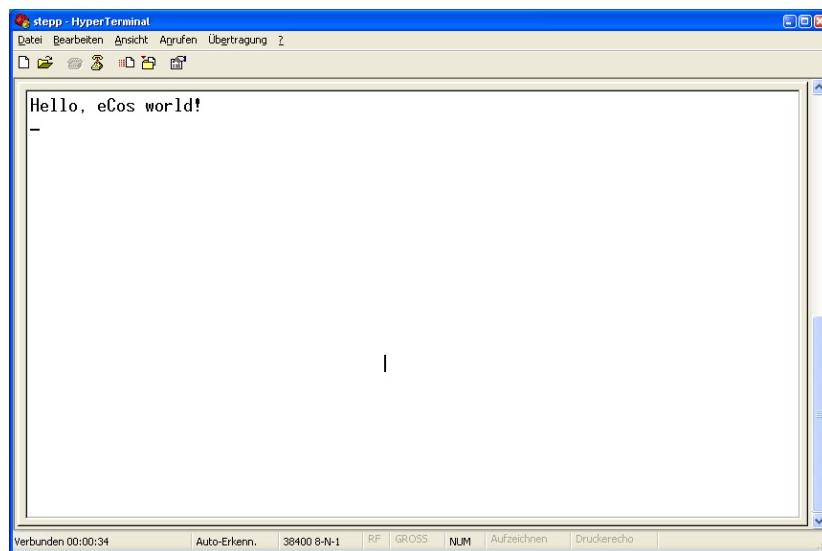
- If an error is generated by clicking the Execute button or during flash programming, check cables and retry the operation or click the help button to get the troubleshooting described in the help file of SirfFlash program.
- If download has been completed successfully, power down the target device. Your target device is ready to run with the project. The target system with the new project is automatically started at the next power up (follow the instructions described on the target device evaluation board, see [1]).

Hint: In order to see effect of downloaded project into the Flash of target device, firstly start the terminal application software (e.g. HyperTerminal program) before you start up the target device. It is located in the following directory,:

- ✓ Go to Start > Program > Accessories > Communication and click the HyperTerminal program.
- ✓ On the appeared screen assign the name for the current connection (e.g. "Falcom device") and click OK.
- ✓ Then choose the correct COM Port to which the module is connected and select the baud rate of (38000 bps, 8 bit, no parity bit, 1 stop bit), then click OK

Set the target device back to the normal operating mode.

Once the target system is started-up it responds a message, which appears on the terminal software screen. See figure below. After each reset of target device the message will be sent via serial port to the terminal software.



A brief description for all included demo applications added in the chapter Appendix [6.1](#).

6 Appendix

6.1 Brief description of demo projects

The included demo projects released from Falcom GmbH demonstrate different features when you download them to the target device, such as the Falcom XF55 and F35-XXL module.

On the followed sections you will find a brief description for each project.

During testing the released projects, an error message will be activated when an error occurs in background processing (for example, failed to connect or failed to send or receive in the background). The message will be displayed on the terminal software screen.

Each project below begins running immediately after power is applied or a reset is triggered on the target device.

6.1.1 Demo Hello

- „dev/ser0“ can not be selected as hardware UART, but as output *diag_printf*.
- via *printf* funktion project sends out the message „Hello, eCos World!“.

6.1.2 Demo Thread

- This project demonstrates how to create and start a thread under Falcom eCos SDK.
- Project initialization a periodic timer (thread) that periodically delivers a message.

6.1.3 Demo Two-thread

- This project shows how to create multiple threads that run simultaneously.
- Each of them delivers a message periodically.

6.1.4 Demo Serial echo

- This project demonstrates how an UART can be accessed (write and read functions are available on the source code).
- The project responses the same character that it receives.

6.1.5 Demo Serial Bridge

- This project shows how the serial ports can be combined.
- The project reads data from the Serial port A and writes (the same) data to the Serial Port B and vice versa.

6.1.6 Demo IOs

- This project demonstrates how to use GPIO's as outputs.
- The effects of this project are shown via LED's. The LED's are continually blinking with specific rhythm (GPIO-Led 1 to GPIO-Led 8 of the target device evaluation board, see [1]).

6.1.7 Demo Flash

- This project demonstrates the write, read and erase functions to the specific sectors of Flash memory.
- The project erases specified sectors of target device's Flash memory at 0x40010000-0x40020000. It also writes a block of data (integer type) into erased Flash memory sectors. After writing has been stopped, project checks the written block of data to Flash (integer type).

6.1.8 Demo Http client

- According to the configuration, project establishes a GPRS data connection. Then by means of the TCP connection a file named „/rfc/rfc0793.txt“ will be downloaded from the Web Server 132.151.6.21.
- Note that the configuration of this project has to be done according to used SIM card and your network operator.

6.1.9 Demo Http Server

- This project demonstrates the target device as a Web server. It establishes a GSM data connection via GSM server (freenet.de) and then starts a HTTP-Server. It has static HTTP resources, including a HTML page stored in the internal Flash of target device.
- An IP-address will be allocated from the remote Web server to the target device. A function included in the source code of project stored in the flash memory reads allocated IP-address and sends it out (IP-address is displayed in a terminal program). By means of the IP-address the HTTP page stored in the internal Flash of target device can be requested from a remote computer via a Web server. Open your Internet browser and enter the IP-address delivered from the target device in the address bar. The project outputs the HTTP page showing following information:

„XF55 now knows TCP/IP!“

„Reloads: 0“

Reload <int> value increments itself after the user updates the uploaded page.

Note that, the size of GSM UART buffer for TCP/IP applications is recommended to be at least 2 Kbyte.

6.2 ISP, GPRS configuration parameters of German service provider

The following table presents GPRS parameters of selected German service providers and operators:

	T-D1	Vodafone D2	E-Plus	Genion O2
APN	internet.t-d1.de	volume.d2gprs.de	internet.eplus.de	APN=internet
REQoS	1,3,4,3,0,0	1,3,4,3,7,31	1,2,4,3,9,31	REQoS=1,2,4,3,9,31
MIQoS	1,0,0,0,0,0	1,0,0,0,0,0	1,0,0,0,0,0	MIQoS=1,0,0,0,0,0
USERNAME	(optional)	""(None)	"eplus"	USERNAME=""(None)
PASSWORD	(optional)	""(None)	""(None)	PASSWORD=""(None)

Table 1: Service provider information, valid 16.10.2001